Assignment: Dynamic Programming & Backtracking

Note: These problems are to be discussed as part of the Group Assignment. (Check this week's Group Assignment on Canvas for details).

The questions asked in this assignment – code implementation and time complexity of your code should be done individually based on the problem-solving strategy discussed within your group.

1. **Solve Dynamic Programming Problem and find its optimal solution.**
   Given a list of numbers, return a subset of non-consecutive numbers in the form of a list that would have the maximum sum.

   Example 1: Input: [7,2,5,8,6]
   Output: [7,5,6] (This will have sum of 18)
   Example 2: Input: [-1, -1, 0]
   Output: [0] (This is the maximum possible sum for this array)
   Example 3: Input: [-1, -1, -10, -34]
   Output: [-1] (This is the maximum possible sum)

   a. Implement the solution of this problem using dynamic Programming. Name your function **max_independent_set(nums)**. Name your file **MaxSet.py**
   b. What is the time complexity of your implementation?

2. **Implement a backtracking algorithm**
   a. Write the implementation to solve the powerset problem discussed in the exercise of the exploration: Backtracking. Name your function **powerset.py**. Name your file **PowerSet.py**
   b. What is the time complexity of your implementation?

`Debriefing (required!): -------------------------`

Report:

Fill the report in the Qualtrics survey, you can access the link here.

(https://oregonstate.qualtrics.com/jfe/form/SV_eKcApyE0Cuoydfg)

`Note: 'Debriefing' section is intended to help us calibrate the assignments.`